

Многоверсионность Изоляция



Авторские права

© Postgres Professional, 2019 год.

Авторы: Егор Рогов, Павел Лузанов

Использование материалов курса

Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

Обратная связь

Отзывы, замечания и предложения направляйте по адресу:

edu@postgrespro.ru

Отказ от ответственности

Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или косвенным, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

Транзакции и их свойства

Стандарт SQL и уровни изоляции

Особенности реализации в PostgreSQL

Управление изоляцией транзакций

Транзакция — множество операций,
которые переводят базу данных из одного
корректного состояния в другое корректное
при условии, что транзакция выполнена полностью
и без помех со стороны других транзакций

Согласованность

Атомарность

Изоляция

атомарность: транзакция либо выполняется полностью,
либо не оставляет никаких следов

согласованность определяется ограничениями целостности
и семантикой приложения

неполная изоляция приводит к нарушению корректности (аномалиям)
при конкурентном выполнении транзакций

Транзакцией называется множество операций, выполняемое приложением, которое переводит базу данных из одного корректного состояния в другое корректное состояние (Согласованность) при условии, что транзакция выполнена полностью (Атомарность) и без помех со стороны других транзакций (Изоляция).

Корректное (согласованное) состояние определяется как ограничения целостности, определенными на уровне базы данных, так и семантикой приложения.

Идеальная изоляция транзакций гарантирует, что на работу приложения не окажут влияния никакие другие конкурентно выполняющиеся транзакции. Однако если ослабить требования к изоляции, то транзакция, которая корректно выполнялась в базе данных в одиночестве, может выдавать некорректные результаты при наличии других транзакций. Такие некорректные ситуации называются аномалиями.

Однако реализация полной изоляции — сложная задача, сопряженная с уменьшением пропускной способности системы. На практике как правило применяется именно ослабленная изоляция, и поэтому важно понимать, какие это влечет последствия.

«Грязное» чтение

запрос видит незафиксированные изменения других транзакций

Неповторяющееся чтение

повторное чтение строки вернет другое значение, если оно было изменено (и зафиксировано) другой транзакцией

Фантомное чтение

повторный запрос по одному и тому же условию вернет иную выборку, если другая транзакция добавила (и зафиксировала) новые строки, удовлетворяющие этому условию

Отсутствие всяких аномалий

результат выполнения транзакций совпадает с результатом последовательного выполнения транзакций

Стандарт SQL допускает четыре уровня изоляции, которые определяются в терминах аномалий, которые допускаются при конкурентном выполнении транзакций на этом уровне:

- **«Грязное» чтение** (dirty read). Транзакция T1 может читать строки измененные, но еще не зафиксированные, транзакцией T2. Отмена изменений (ROLLBACK) в T2 приведет к тому, что T1 прочитает данные, которых никогда не существовало.
- **Неповторяющееся чтение** (non-repeatable read). После того, как транзакция T1 прочитала строку, транзакция T2 изменила или удалила эту строку и зафиксировала изменения (COMMIT). При повторном чтении этой же строки транзакция T1 видит, что строка изменена или удалена.
- **Фантомное чтение** (phantom read). Транзакция T1 прочитала набор строк по некоторому условию. Затем транзакция T2 добавила строки, также удовлетворяющие этому условию. Если транзакция T1 повторит запрос, она получит другую выборку строк.

На самом деле существуют и другие аномалии, но они не отражены в стандарте.

Полное отсутствие каких бы то ни было аномалий достигается, если команды, выполняемые в конкурентно работающих транзакциях, приводят к такому же результату, какой получился бы в случае последовательного — одна завершилась, следующая началась — выполнения этих транзакций (при каком-нибудь порядке выполнения).

	«грязное» чтение	неповторяющееся чтение	фантомное чтение	другие аномалии
Read Uncommitted	да	да	да	да
Read Committed	—	да	да	да
Repeatable Read	—	—	да	да
Serializable	—	—	—	—

на всех уровнях не допускается потеря зафиксированных изменений

В стандарте SQL определены четыре уровня изоляции транзакций: Read Uncommitted, Read Committed, Repeatable Read, Serializable (по умолчанию).

Таблица показывает соответствие между уровнями изоляции и теми аномалиями, которые они могут допускать.

Обратите внимание, что для каждого из четырех уровней изоляции стандартом SQL не допускается потеря уже сделанных и зафиксированных изменений в других транзакциях (lost updates).

Также нужно отметить, что для обеспечения уровня изоляции Serializable недостаточно исключить три аномалии, определенные в стандарте.

	«грязное» чтение	неповторяющееся чтение	фантомное чтение	другие аномалии
Read Uncommitted	—	да	да	да
Read Committed	—	да	да	да
Repeatable Read	—	—	—	да
Serializable	—	—	—	—

на всех уровнях не допускается потеря зафиксированных изменений

6

Реализация уровней изоляции в PostgreSQL строже, чем требуется стандартом:

- благодаря многоверсионности (MVCC) «грязное» чтение не допускается вообще. Этот уровень не стоило бы использовать, даже если бы он поддерживался, поскольку он позволяет увидеть то, чего никогда не было.
- уровень Repeatable Read не допускает не только неповторяемое чтение, но и фантомное чтение (хотя и не обеспечивает полную изоляцию).

Уровнем изоляции по умолчанию является Read Committed.

<https://postgrespro.ru/docs/postgresql/10/transaction-iso>



Read Uncommitted не поддерживается

«грязное» чтение невозможно ни на одном уровне

Read Committed используется по умолчанию

допускаются всевозможные аномалии,
включая неповторяемое чтение и фантомное чтение

Repeatable Read строже, чем требует стандарт SQL

не допускает фантомного чтения

Serializable гарантирует полную изоляцию

1. Создайте функцию, возвращающую значение «1».
В другом сеансе начните транзакцию с уровнем изоляции Read Committed и вызовите функцию.
Теперь измените функцию так, чтобы она возвращала «2».
Какое значение вернет функция, если снова вызвать ее в открытой ранее транзакции?
Объясните результат.
2. Создайте пустую таблицу.
Откройте транзакцию с уровнем изоляции Repeatable Read (и пока не выполняйте в ней никаких команд).
В другом сеансе добавьте в таблицу строку.
Видна ли добавленная строка в открытой транзакции?

1. Функцию можно создать (и изменять) на основе такого шаблона:

```
CREATE OR REPLACE FUNCTION f() RETURNS integer AS $$  
SELECT 1;  
$$ LANGUAGE sql;
```